Performance Analysis and Consolidation Engineering (*PACE)*

Air Force Research Laboratory

Information Directorate

Information and Intelligence Exploitation Division

Intelligence Data Handling Branch

AFRL/IFEB

John Vona, Branch Chief, vonaj@rl.af.mil, (315)-330-3601

Anthony Macera, PACE Project Manager, maceraa@rl.af.mil, (315)-330-2321

http://www.if.afrl.af.mil/tech/programs/pace/

Why PACE?

- Reduce O&M costs
  - More cost-effective use of computing resources
    - Accurately determine platform configurations that meet performance requirements.
    - Plan system capacity to meet future performance demands.
    - Reduce hardware costs with fewer deployed servers.
    - Establish a "Buying Guide" for site planning to accommodate new and upgraded applications
  - Reduce costs for software licenses.
- Develop systems engineering processes for USIGS integration
  - Start early in USIGS life cycle to create the right "culture" for development
  - Apply state-of-the-art commercial practices and tools to insure quality and cost-effective development practices

Why PACE?

- Reduce costs for fail-over back-up systems
  - Consolidated server can provide back up for multiple systems
- Move toward elimination of overlapping application functionality
  - Consolidated applications can share data, services, and support applications

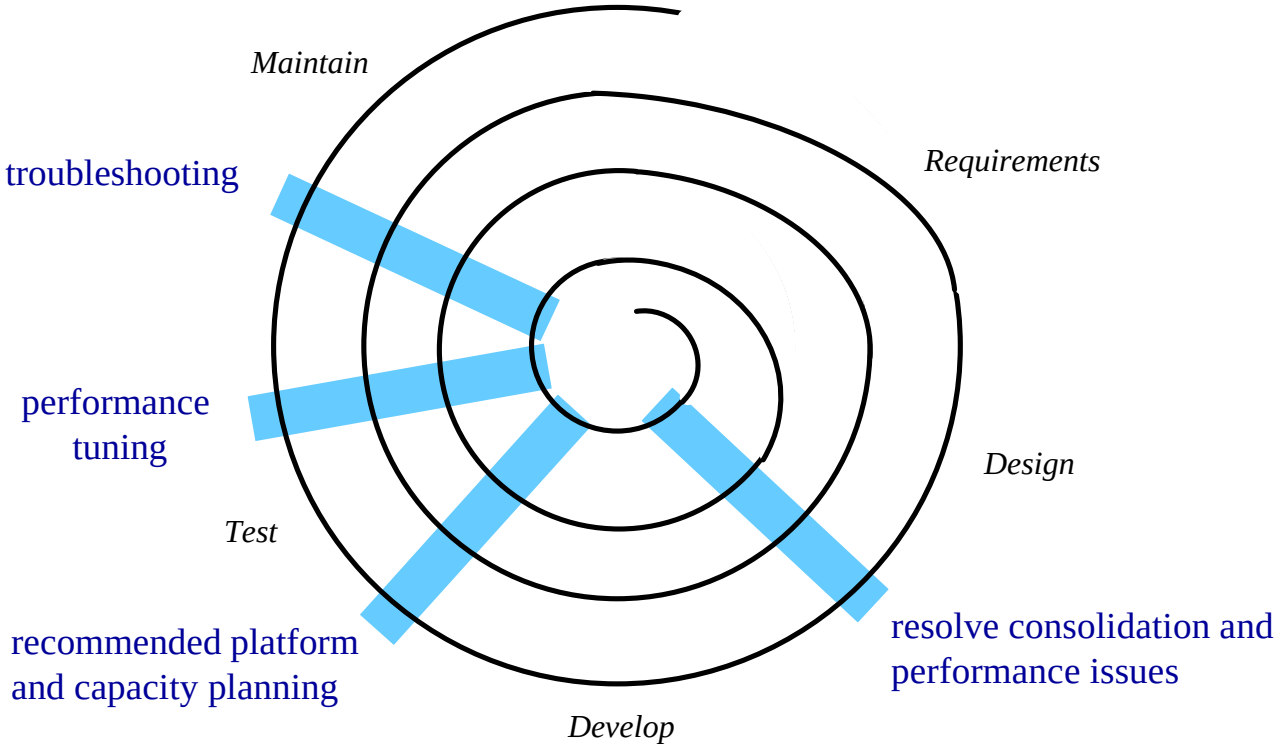
What is PACE?

- Methodology
  - documented approach for conducting performance analyses
- Capacity Planning
  - plan for anticipated load and usage changes
- Tuning and Troubleshooting
  - identify individual application issues and make platform changes to improve performance
- Tools
  - use sophisticated commercial products and customized publicly available software

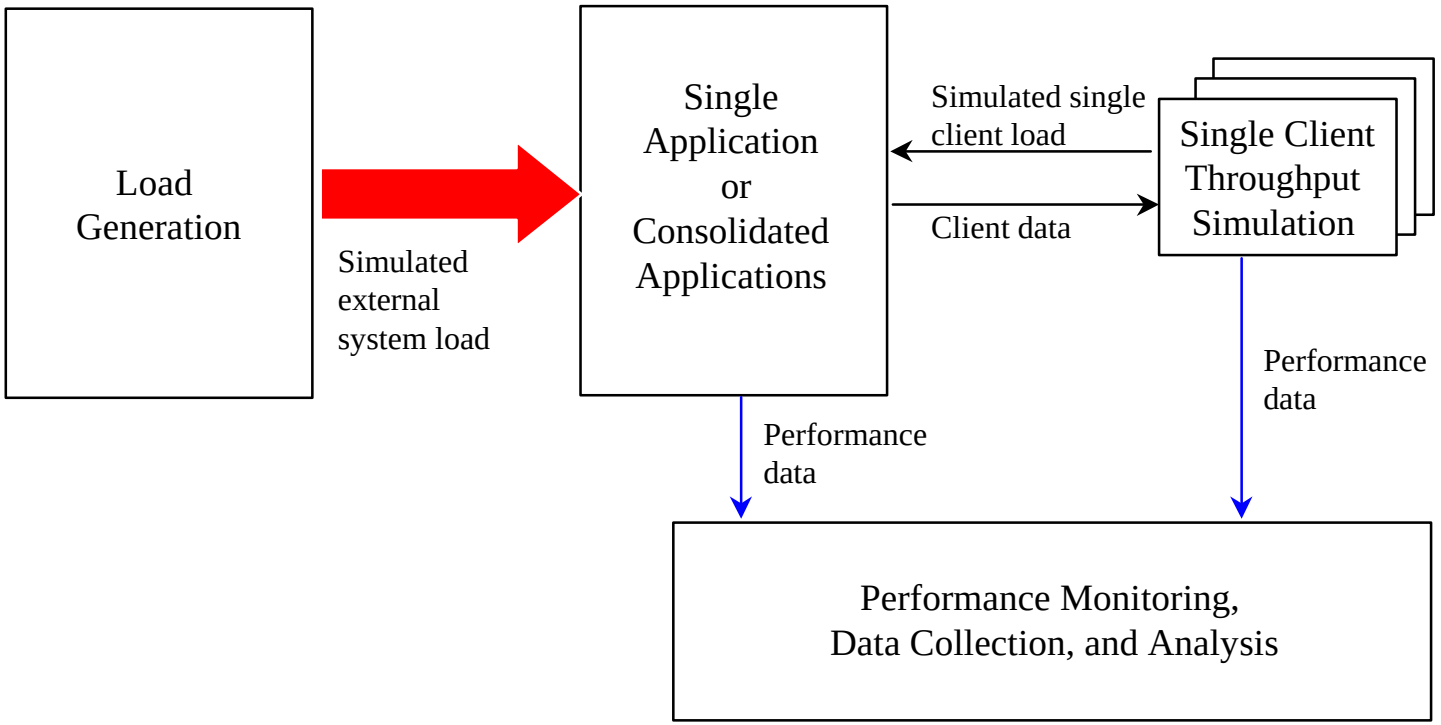
What is PACE?

- Consolidated Installation Procedures
  - reconcile installation procedures for convenient installation of consolidated applications
- Consolidated Database Services
  - determine database server configuration to support multiple applications
- Version Reconciliation
  - determine version compatibility for supporting software to avoid redundancy
- Buying Guide
  - catalogue of configurations and performance characteristics

PACE in the Application Life Cycle



Conceptual Design for Performance Analysis



PACE Methodology

Iterative 5-step Process for Single and Consolidated Application Platforms

- Application Installation
- Application Characterization
- Load Generation
- Testing
- Analysis

Application Installation

- Installation as per application documentation.
- Understanding required administrative tasks; e.g., setting up user accounts, audit management backups.
- Integration onto consolidated platform - after individual application characterization.

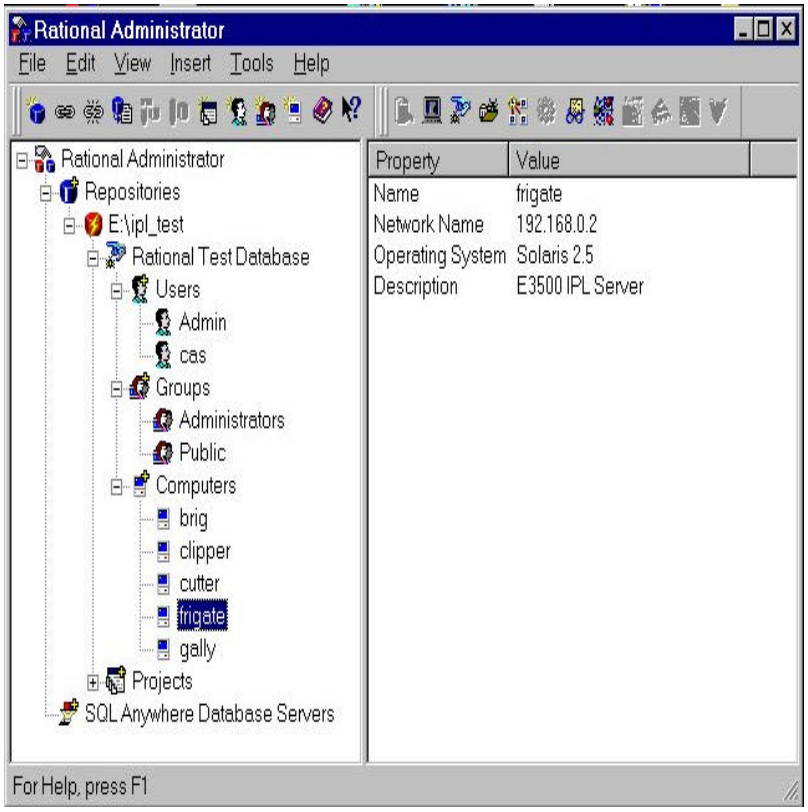
Application Characterization

- Identify processes, COTS utilization; e.g., SYBASE
- Determine profile; e.g. memory usage, I/O requirements, network utilization
- Establish usage scenarios; how is application typically used at sites
- Variety of software tools to examine application in quiescent and active states

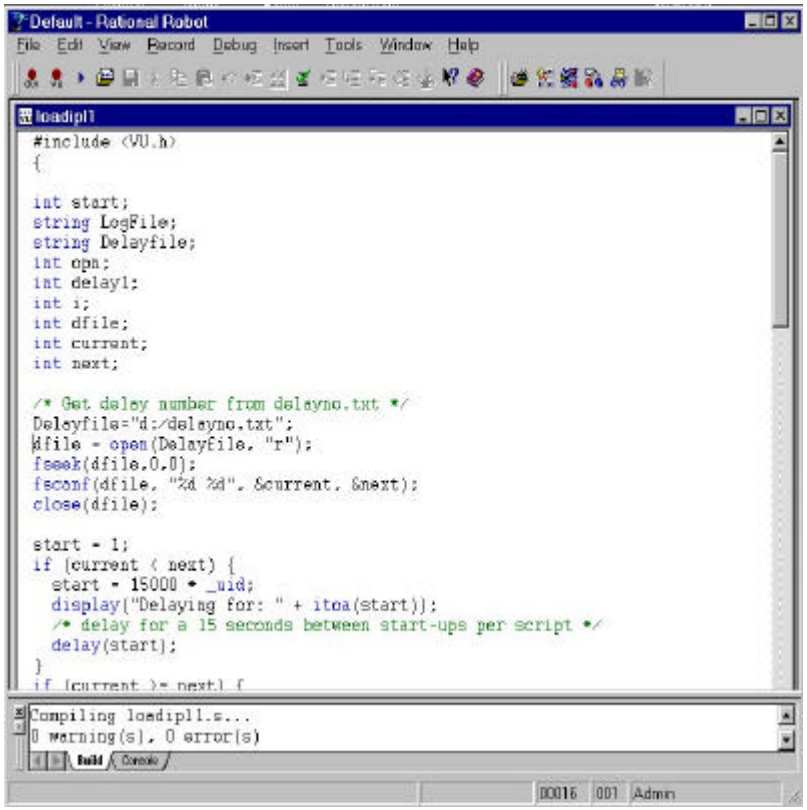
Load Generation

- Simulate real-world usage by recording actual usage and replaying in test laboratory.
- Scaling simulation for in-the-field conditions; e.g., number of users, hw and sw configurations.
- Create multiple application loading scenarios; e.g., heavy, medium, light.
- Rational's Performance Studio automates much of the processing

Using Performance Studio

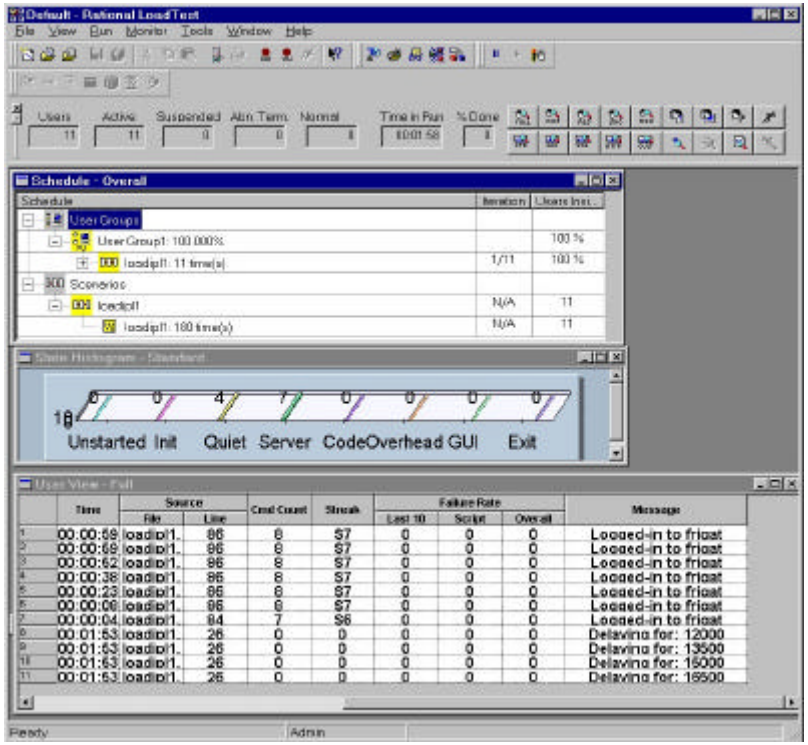


1. Create a Rational repository in Rational Administrator to store test assets.

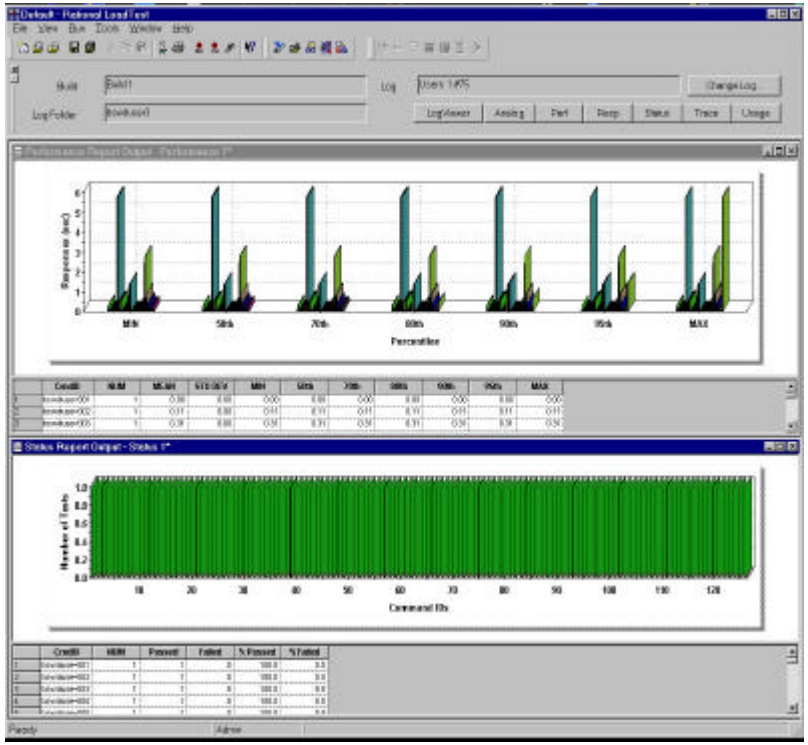


2. Record scripts in Rational Robot that emulate client/server conversations.

Using Performance Studio



3. Create schedules in rational Loadtest that emulate client sending requests to a server



4. Run and monitor the schedules as they add load to your database and Web servers.

Testing

- Run varying load scenarios and collect performance data
- Monitor performance features during testing
- Test runs are automated using Rational Performance Studio
- Performance metric data is recorded by Landmark's PerformanceWorks, SE Toolkit, Ethereal, Memtool, Proctool, and Strace

Analysis

- Analyze client throughput looking for anamolies and constraints
- Compare performance data with recommended guidelines and expected behavior
- Compare performance data before/after configuration changes and tuning
- Use capacity planning models to improve performance, implement results of models where possible

Analysis for Performance Tuning

- OS configuration; e.g., priority paging, dynamic load balancing
- DBMS configuration, e.g. query caching
- Network topology and configuration, e.g., buffer sizes for network subsystems
- Hardware recommendations, e.g., more or faster processors

Analysis for Performance Troubleshooting

- Network problems, e.g., high collision or re-transmission rates
- I/O bottlenecks, e.g., high activity on single disk
- High context switching or mutex locking (high kernel resource contention and/or inefficient use of multiple CPU's)

The Case for Capacity Planning

UNIT RIG
A DIVISION OF TEREX CORPORATION

Jeep® Cherokee

New Unit Rig MT5500 AC Drive Haul Truck

To scale

VEHICLE	Jeep®Cherokee	MT5500
gross weight - lb. (kg)	4528 (2054)	1,125,000 (510 204)
length - ft (m)	14.1 (4.3)	46 (14)
width - ft (m)	5.88 (1.79)	30 (9.1)
height - ft (m)	5.28 (1.61)	24 (7.3)
payload - short ton (metric ton)	.75 (0.68)	340 (308.43)
top speed – mph (kph)	100 (161.3)	40 (64.5)
TIRE		
cost - US$	75	27,500
height - ft (m)	2.28 (0.7)	12.8 (3.90)
weight - lb (kg)	27 (12.25)	9750 (4422)
ENGINE		
power - hp (kW)	190 (142)	3000 (2238)
fuel consumption - gal/hr (L/min)	3.33 (0.21)	136.8 (8.63)
displacement - cubic inch (L)	244.1 (4.0)	3967 (65)
oil capacity - qt (L)	6 (5.68)	264 (250)
dry weight - lb (kg)	500 (227)	15813 (7173)

Jeep is a registered trademark of the Chrysler Corp

Capacity Planning

- Use data collected during performance testing to model system performance
- Allows “what if” analysis for relating hardware and software configuration features to anticipated load
- Landmark Predictor provides tools for data collection, system modeling, and analysis

PACE Software Tools

- Rational Performance Studio
- Landmark PerformanceWorks and Predictor
- Other Tools
  - SE Toolkit (kernel performance metrics)
  - MemTool (memory utilization)
  - ProcTool (process information)
  - Ethereal (network analysis)
  - Strace (system call traces)
- Solaris built-ins

PACE Hardware

- Sun Enterprise 5500, 8 CPU, 8 Gbyte RAM, 128 Gbyte storage array
- Sun Enterprise 3500, 4 CPU, 5 Gbyte RAM, 128 Gbyte storage array
- Sun Enterprise 3000, 4 CPU, 2 Gbyte RAM
- Network Appliance file server, 32 Gbytes
- 2 Sun Ultra-10 workstations, 512 Mb and 1 Gbyte RAM
- 1 Sun Ultra-5 workstation, 128 Mb RAM.
- 2 dual-processor 450 Mhz Pentium II PCs, 512 Mb and 1 Gbyte RAM
  100 Mbit/sec EtherSwitch (upgrading to ATM).

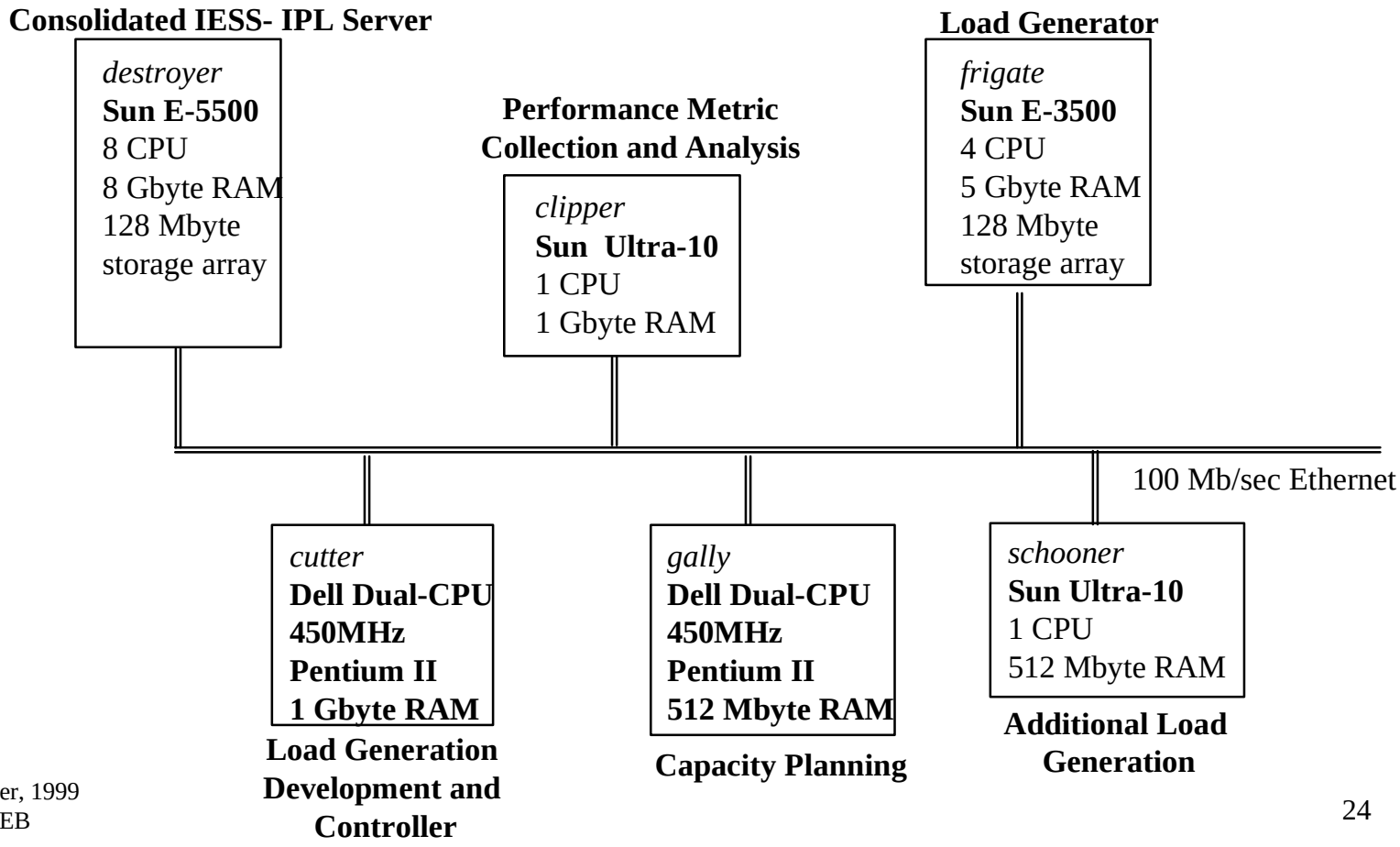
PACE Accomplishments

- MIDB, DAWS, IPL, 5D, RAAP consolidation for USCENTCOM
  - installed on 3 E-4000 servers at USCENTCOM
- Unclassified consolidated application server for testing analysis procedures. (Uses RAAP, 5D and IPL)

Current Efforts

- IESS-RMS consolidation started 9/98 but RMS installation at AFRL/IFEB cancelled by NIMA/SOM 12/98
- Conducting independent IESS - IPL consolidation for AFDCGS
  - Consolidation onto an E-5500 Solaris 2.6 platform
  - Currently completing IPL characterization
- Support for Information Assurance Automated Intrusion Detection Environment (IA:AIDE) analysis
  - Capacity planning for Oracle database

PACE Architecture for NIMA CAS



IPL Performance Analysis

- IPL (Image Product Library) provides a repository of image products where items may be queried and delivered either electronically or via conventional means
- IPL analysis was undertaken in order to consolidate with IESS
- IPL analysis allowed refinement of performance analysis methods and supporting tools

IPL Analysis Approach

- IPL 2.01 installed on E-3500 with 4 processors and 5 Gb RAM, 128 Gbyte drive array
- Broadsword Gatekeeper installed on separate machine so IPL performance could be isolated.
- Tests of product ingestion and small and large client queries conducted using 366-1000 products
- Automated testing and measurement tools used to simulate user interaction and collect and analyze performance metrics

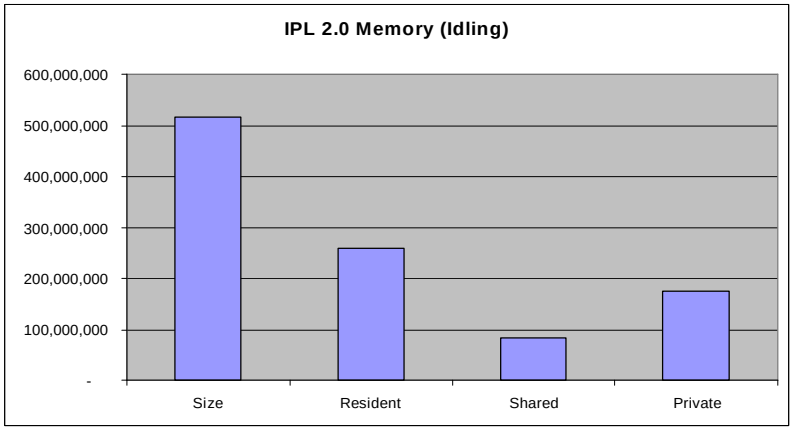
IPL Installation Notes

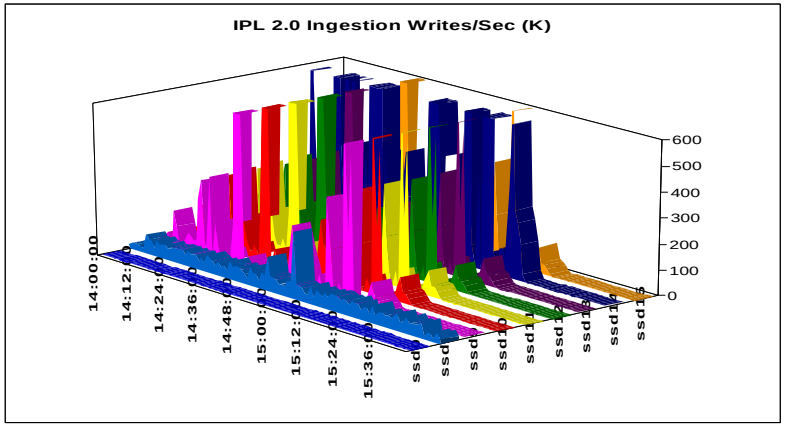
- Could not use current Sun recommended patches
- Did not use recommended shell for ‘dummy2’ user (/bin/FALSE)
- Auditing flags were not enabled and Client Server Environment (CSE) was not installed
- DNS used for host name lookup
- All data except for database devices were on RAID volumes

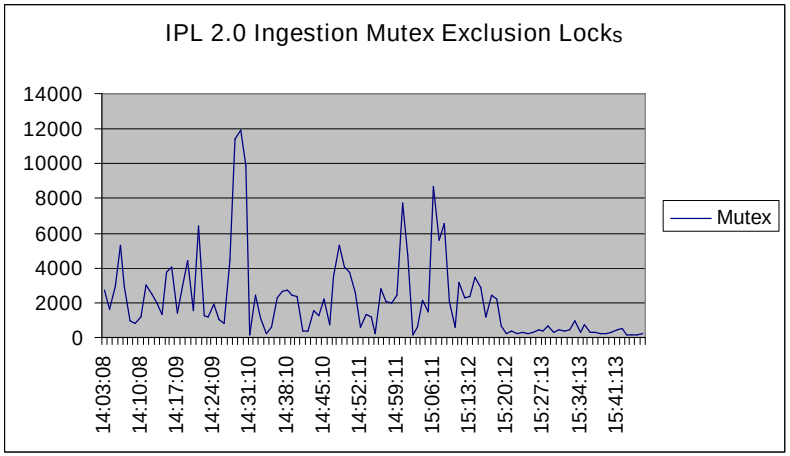
IPL Characterization

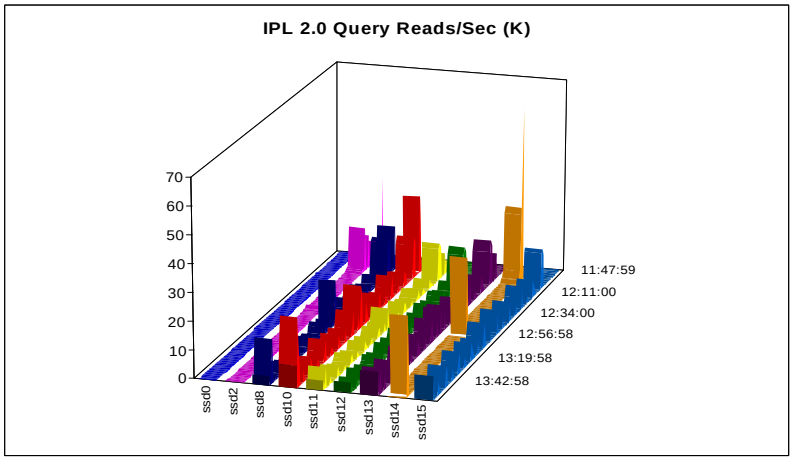
- Two main operating modes
  - Ingestion instantiates products one at a time
  - Querying is multithreaded
- Exorbitant use of network ports makes consolidation difficult and hinders uninterrupted operation
- Memory requirements for acceptable operational performance conflict with vendor configurations
- Ingestion exhibits high kernel resource contention
- Heavy dependence on disk I/O during queries mandates proper load balancing to ensure adequate performance

IPL Characterization









IPL Load Generation

- Four Test Cases
  - Ingestion. Broadsword on separate server (LDK5)
  - Query (large/small). Broadsword on same server (LDK6)
  - Ingestion/Query. Broadsword on separate server (LDK7)
  - 11 concurrent user query test. Broadsword was replaced by Performance Studio (R11)

IPL Load Generation

- Initial ingestion tests were manual copy of ~125 image products into the auto-ingest directory
- Queries were single user and returned 237 records (large) and 97 records (small)
- Estimates from SEToolkit and Predictor
  - scaled CPU use as reported by vmstat.se => 8 users possible
  - scaled CPU use as reported by device utilization analysis report => 28 users possible
  - repeated trials indicated maximum of 11 users possible.
- The 11 user test used a 63 record query,

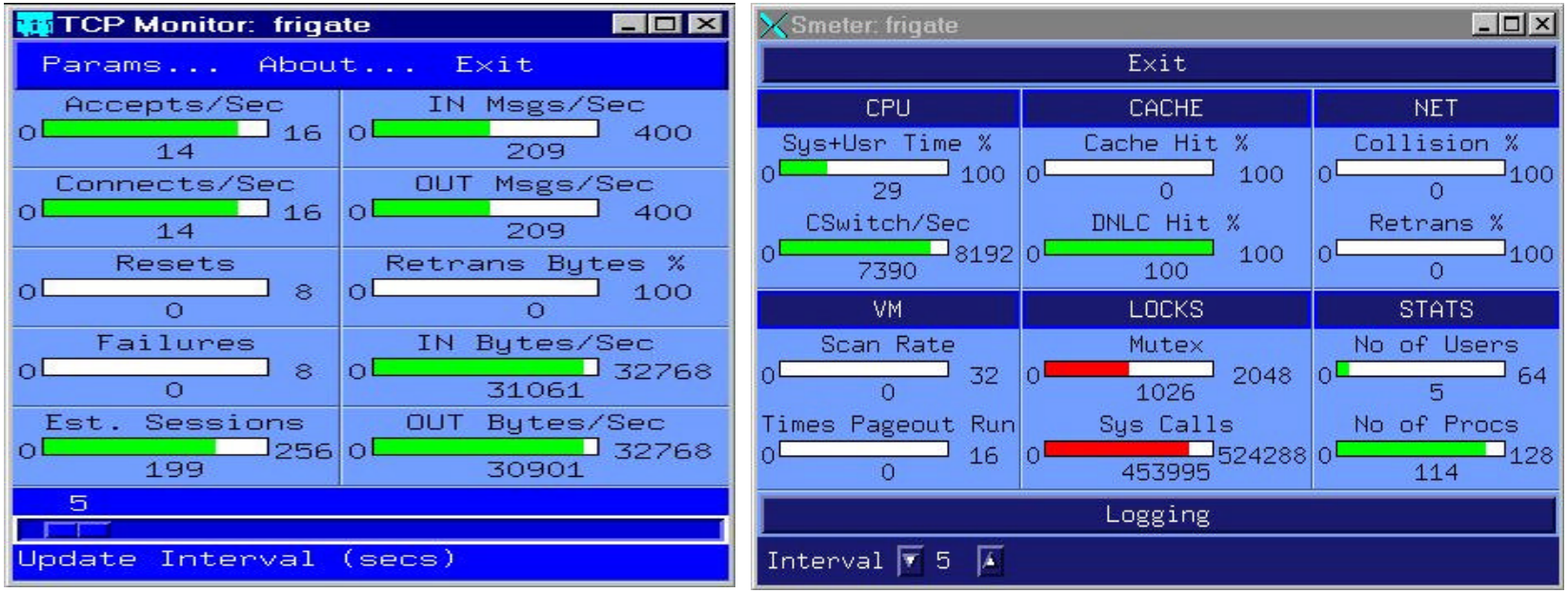
IPL Testing

- Performance Studio generates load using GUI or virtual users
- SEToolkit, Unix Smart Agent (SA), Memtool, Ethereal, and Proctool record and display data at 60 sec intervals
- Data is cross-checked between SEToolkit and Unix SA

IPL Testing

- SEToolkit was also used to monitor machine under test at higher frequencies to view transient events
- Processor, process, disk activity recorded by Unix SA for capacity planning
- Data recorded in ASCII files for input to Excel

SEToolkit Real-time Monitoring

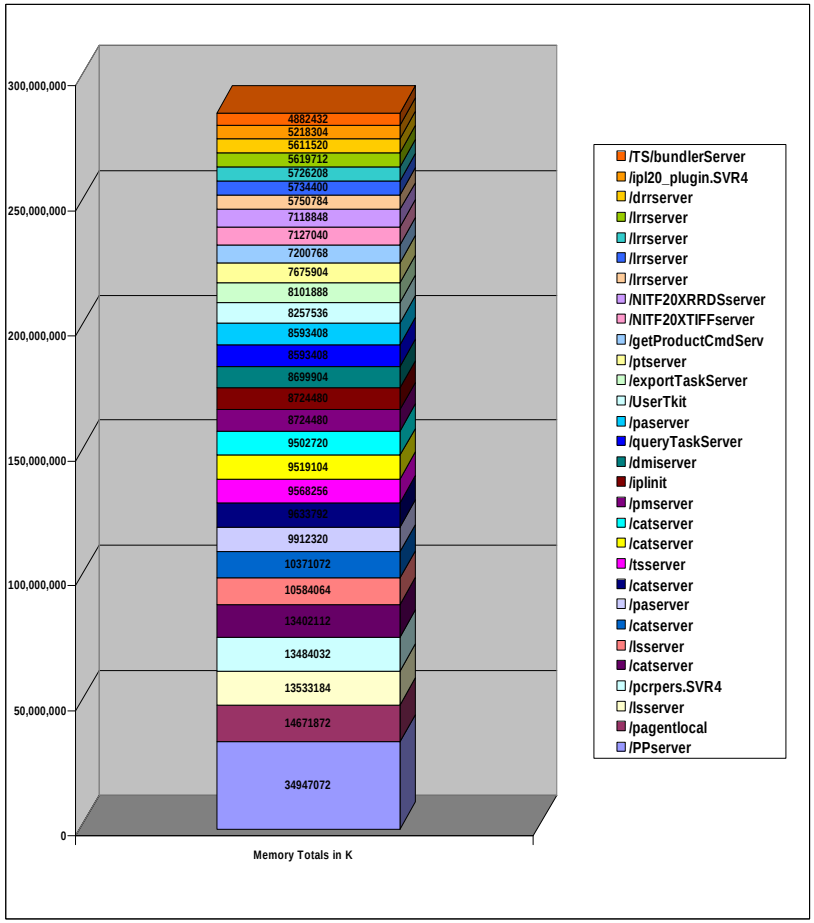


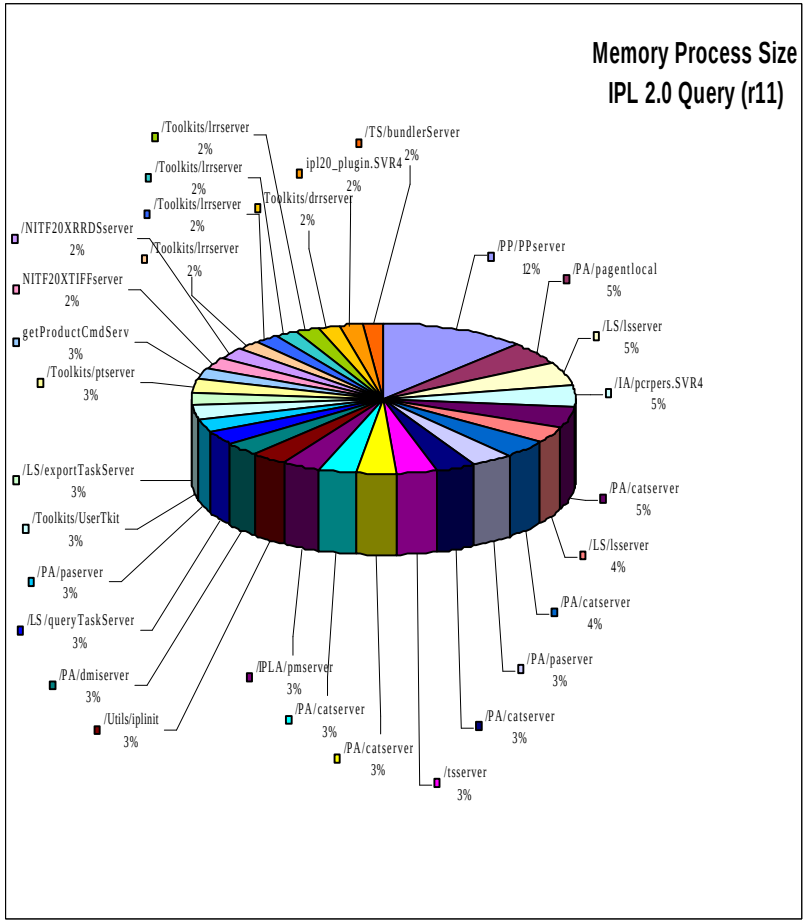
IPL running in a quiescent mode

IPL Analysis

- Results/Conclusions are based on test cases described
- Erratic IPL behavior made testing difficult
- Results varied when tests were repeated
- Monte Carlo analysis was not possible
- IPL's greatest encumbrance is CPU servicing of the network driver
- Memory use under peak load conditions (R11) was ~3 Gb and memory was not reclaimed

IPL Memory Usage

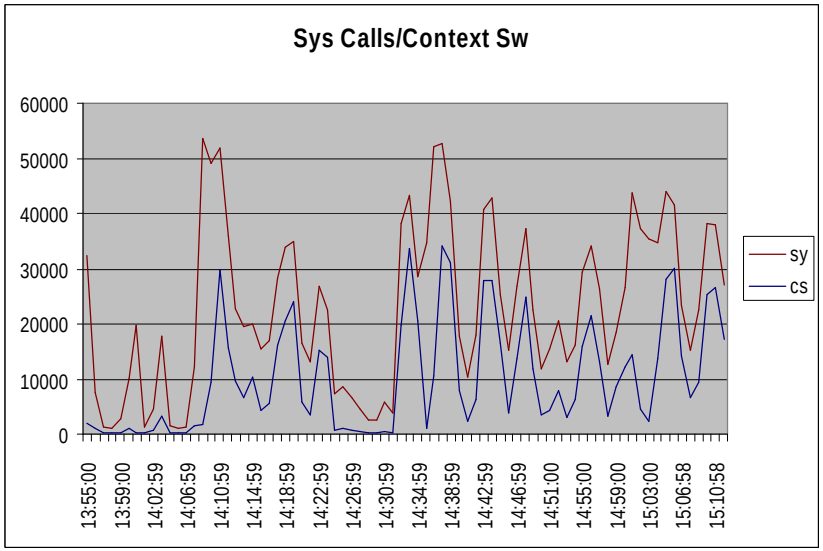


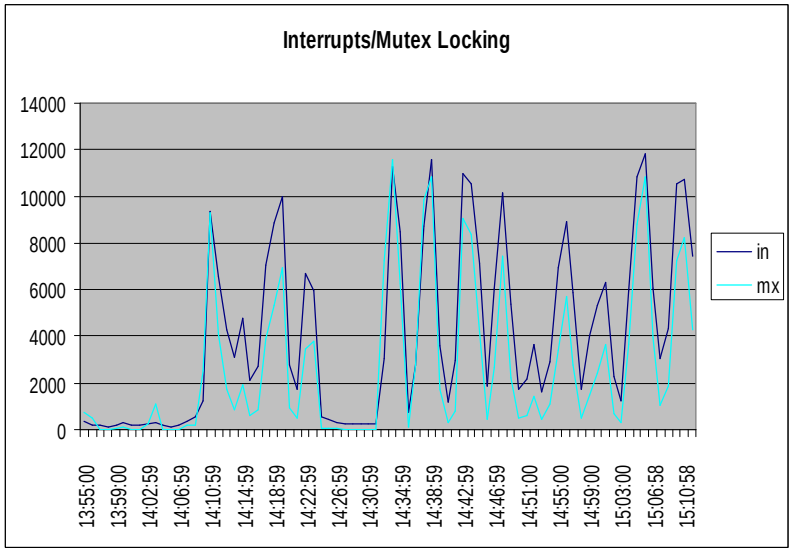


IPL Analysis

- Number of I/O ports and connections in use was high; ~125 ports idling, ~700 ports processing
- Ingestion is predominately single threaded and heavily I/O bound
- Queries have a propensity to expend I/O resources and require large amounts of memory
- All test activity showed high mutex locking with attendant system calls and context switching

IPL Mutex Locks/Context Switches





System Calls: Number of calls to system provided services by all processes.

Context Switching: Number of Involuntary and voluntary process cpu/memory switching

Interrupts:Number of events that temporarily divert execution to service interrupting device/application.

Mutex Exclusion Locks: 200/sec very busy, > 500/sec warning condition

IPL Analysis - Capacity Planning

- Landmark Systems Predictor used for capacity planning
  - Transfer function is an analytic queuing model with >90% accuracy
  - Algorithms tuned for steady-state heavy load conditions
  - Data from Unix SA averaged into 15 minute intervals
  - Models describe average system behavior in order to predict service levels and resource usage
  - Baseline model of system activity created first using Unix SA data

IPL Analysis - Capacity Planning

- Analysis of R11 data, 11 concurrent users continuously doing a 63 record query
- Workload components defined by process for IPL, Sybase, and Remainder
- View Data module revealed the following in the averaged Unix SA data
  - Physical & logical page inputs/outputs are 0.0 for all intervals
  - Number of free swap pages constant for all intervals
  - %CPU use of (87, 82, 100, 100, 84, 100, 100, 100), threshold is 95%
  - %CPU I/O wait < 5, all intervals, threshold is 5%
  - %CPU Idle < 25, all intervals, threshold is 25%
  - %CPU use was 10 - 17 system, 68 - 88 user
  - %Disk use <= 13, all disks, all intervals, threshold is 30%
  - SSD8,12,15 had high wait times (1000 - 3000 ms), other disks had 0.0

IPL Analysis - Capacity Planning

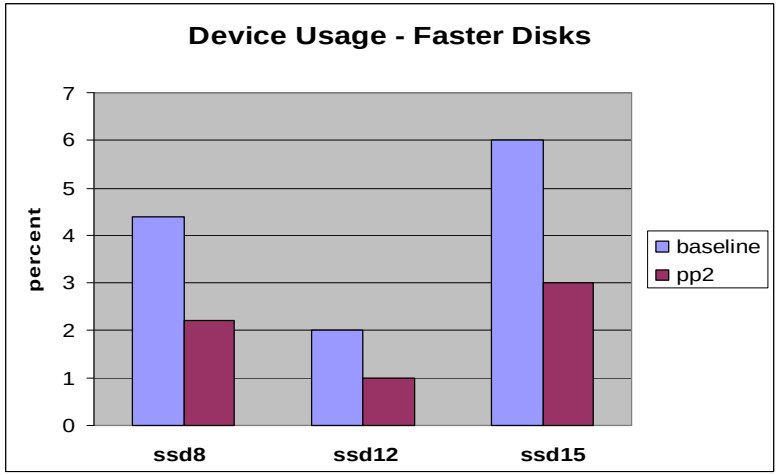
Projection Points (PP) defined from View Data output

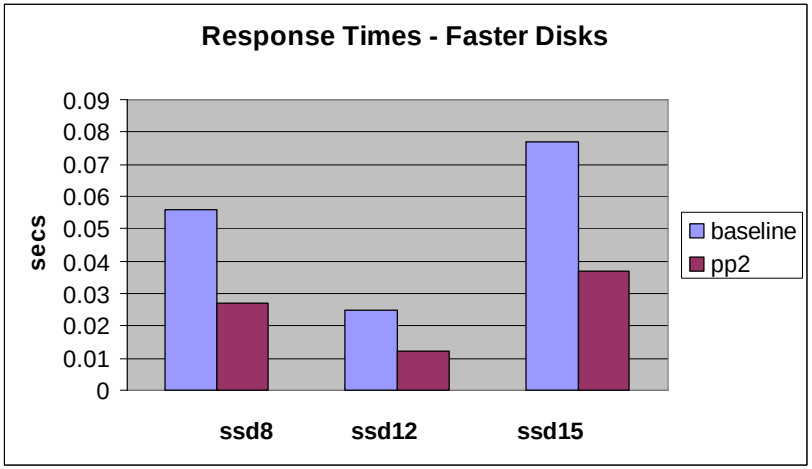
Projection Point (What If)	***Implementation of Result***
CPU upgrade, add four CPUs, same speed as originals (400 MHz) (PP1)	Add CPU cards to chassis if expansion is possible
Disk upgrade for SSD8, 12, 15, use 1/2 of original service times (PP2)	Obtain faster disks if possible and/or add more disks
Redistribute workload on SSD8, 12, 15 with SSD11, 14 based on response times (PP3)	Move files and/or applications among existing disks and/or add more disks

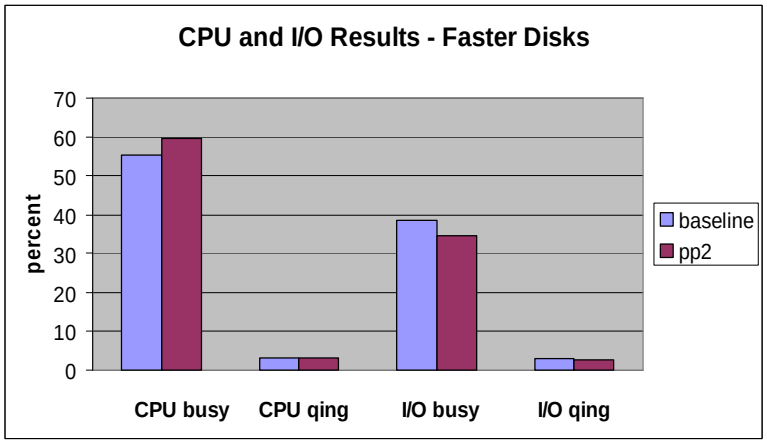
Source of Graphed Results

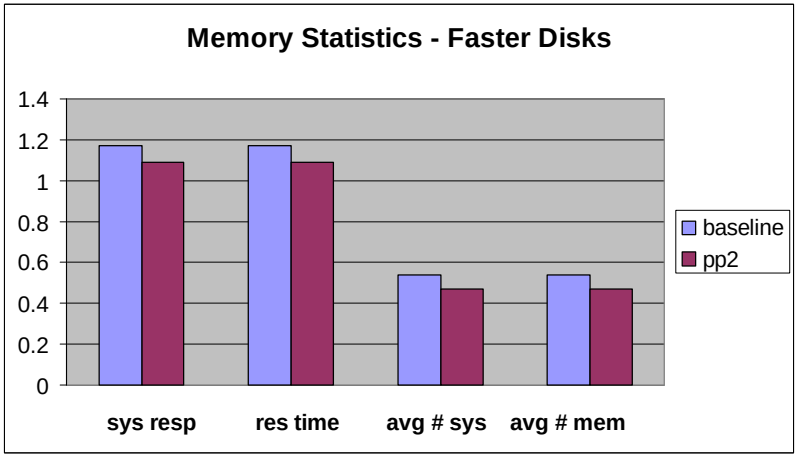
- Remainder Workload Component does actual I/O transfers and has maximum effect on CPU utilization
- IPL Sybase Workload Component had minimal effect on I/O and CPU
- IPL Workload Component, excluding system calls, had minimal effect on I/O and CPU
- Results reported for Remainder Component

Results for PP2 - Faster Disks

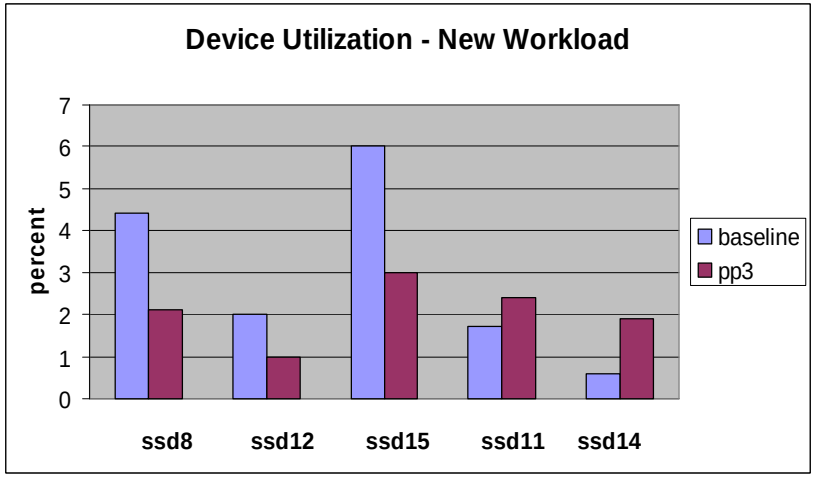


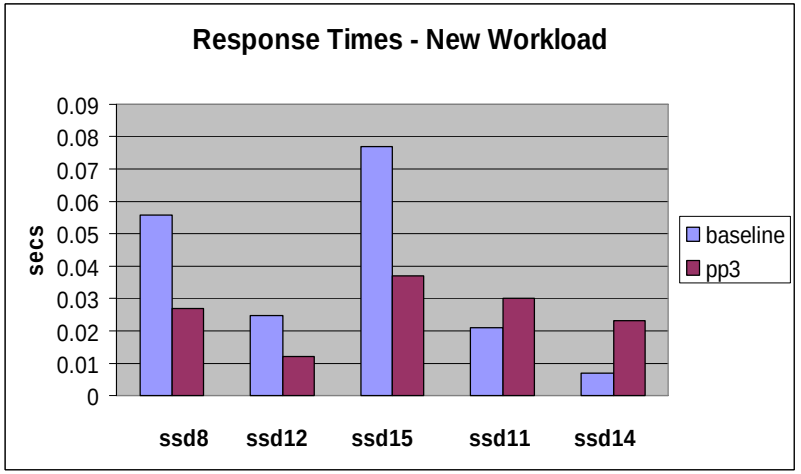


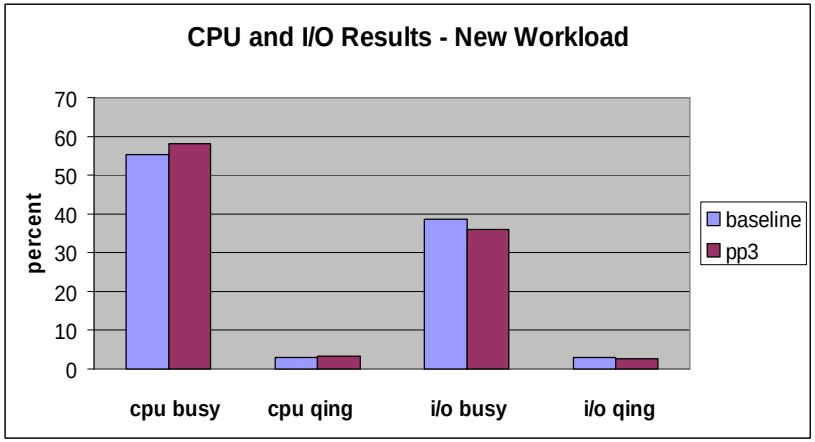


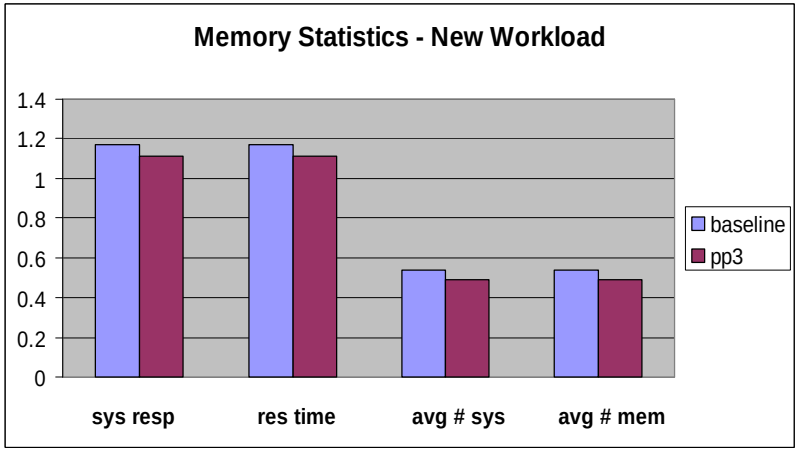


Results for PP3 - New Workload









Summary

- PACE provides the means for more cost-effective application deployment
  - Immediate savings on hardware/software O&M costs
  - Eventual savings on redundant application functionality, system administration staff
- PACE enhances software quality
  - Identify and resolve performance issues during design and development
  - Insure that anticipated performance demands can be met
  - Insure that desired application consolidation is viable
  - Provides a method to record and analyze performance metrics
  - Allows for capacity planning to predict system service levels and resource usage.